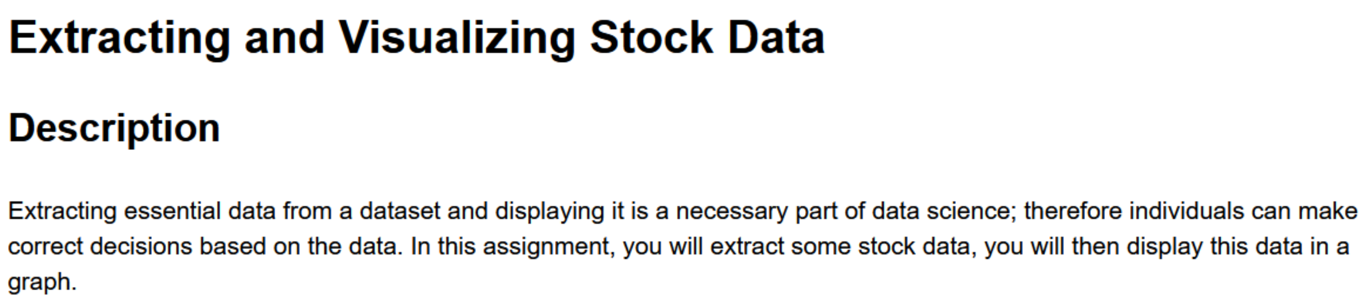
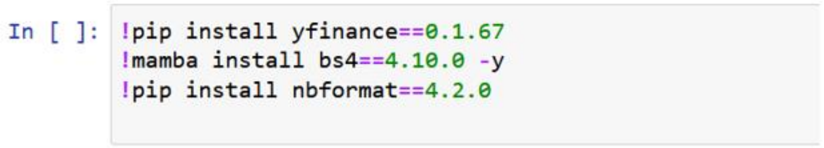


Extracting and Visualizing Stock Data

Description

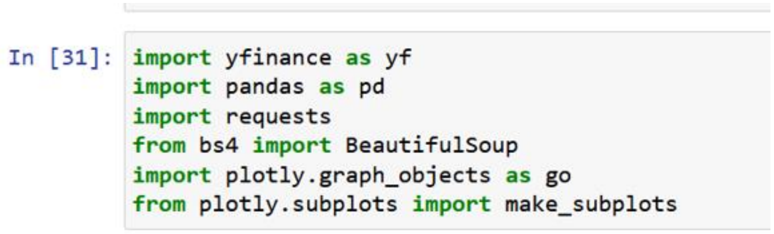
Extracting essential data from a dataset and displaying it is a necessary part of data science; therefore individuals can make correct decisions based on the data. In this assignment, you will extract some stock data, you will then display this data in a graph.

ANSWERS BY GM

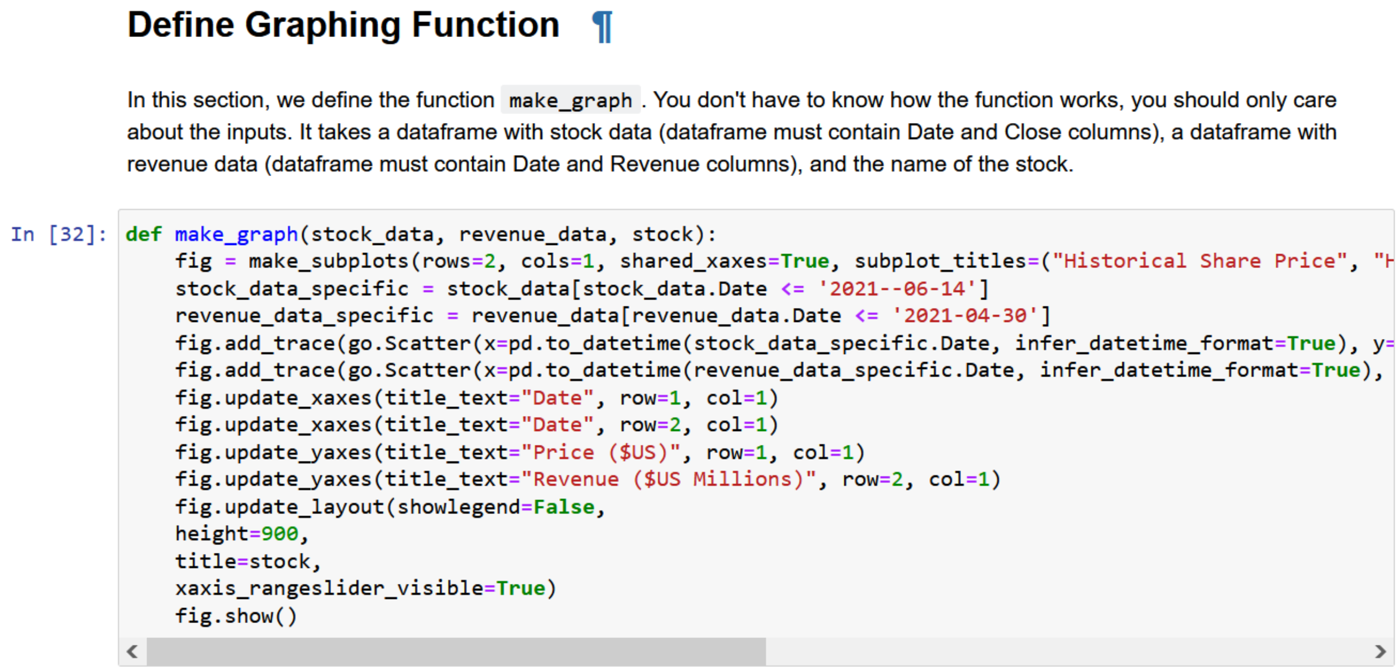


```
In [ ]: !pip install yfinance==0.1.67
        !mamba install bs4==4.10.0 -y
        !pip install nbformat==4.2.0
```

!pip install pandas <-- needed to add this to the lab



```
In [31]: import yfinance as yf
         import pandas as pd
         import requests
         from bs4 import BeautifulSoup
         import plotly.graph_objects as go
         from plotly.subplots import make_subplots
```



Define Graphing Function ¶

In this section, we define the function `make_graph`. You don't have to know how the function works, you should only care about the inputs. It takes a dataframe with stock data (dataframe must contain Date and Close columns), a dataframe with revenue data (dataframe must contain Date and Revenue columns), and the name of the stock.

```
In [32]: def make_graph(stock_data, revenue_data, stock):
             fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "H
             stock_data_specific = stock_data[stock_data.Date <= '2021--06-14']
             revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-30']
             fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date, infer_datetime_format=True), y=
             fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True),
             fig.update_xaxes(title_text="Date", row=1, col=1)
             fig.update_xaxes(title_text="Date", row=2, col=1)
             fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
             fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
             fig.update_layout(showlegend=False,
             height=900,
             title=stock,
             xaxis_rangeslider_visible=True)
             fig.show()
```

Question 1: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is Tesla and its ticker symbol is `TSLA`.

n [114]:

```
tesla_ticker = yf.Ticker("TSLA")
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `tesla_data`. Set the `period` parameter to `max` so we get information for the maximum amount of time.

In [ ]:

```
 tesla_data = tesla_ticker.history(period="max")
```

Reset the index using the `reset_index(inplace=True)` function on the tesla_data DataFrame and display the first five rows of the `tesla_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

In [ ]:

```
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

Question 2: Use Webscraping to Extract Tesla Revenue Data

Use the `requests` library to download the webpage https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm Save the text of the response as a variable named `html_data`.

In [36]:

```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0
html_data = requests.get(url).text
```

Parse the html data using `beautiful_soup`.

In [81]:

```
soup = BeautifulSoup(html_data, 'lxml')
```

Using `BeautifulSoup` or the `read_html` function extract the table with `Tesla Quarterly Revenue` and store it into a dataframe named `tesla_revenue`. The dataframe should have columns `Date` and `Revenue`.

Click here if you need help locating the table

In [ ]:

```
 tesla_revenue = pd.DataFrame(columns=["Date", "Revenue"])

for row in soup.find_all("tbody")[1].find_all("tr"):
    col = row.find_all("td")
    Date = col[0].text
    Revenue = col[1].text
    tesla_revenue = tesla_revenue.append({"Date":Date, "Revenue":Revenue}, ignore_index=True)
```

Execute the following line to remove the comma and dollar sign from the `Revenue` column.

In [ ]:

```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

Execute the following lines to remove an null or empty strings in the Revenue column.

```
In [97]: tesla_revenue.dropna(inplace=True)

         tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
```

Display the last 5 row of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
In [98]: tesla_revenue.tail()
```

Out[98]:

	Date	Revenue
48	2010-09-30	31
49	2010-06-30	28
50	2010-03-31	21
52	2009-09-30	46
53	2009-06-30	27

Question 3: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is GameStop and its ticker symbol is `GME`.

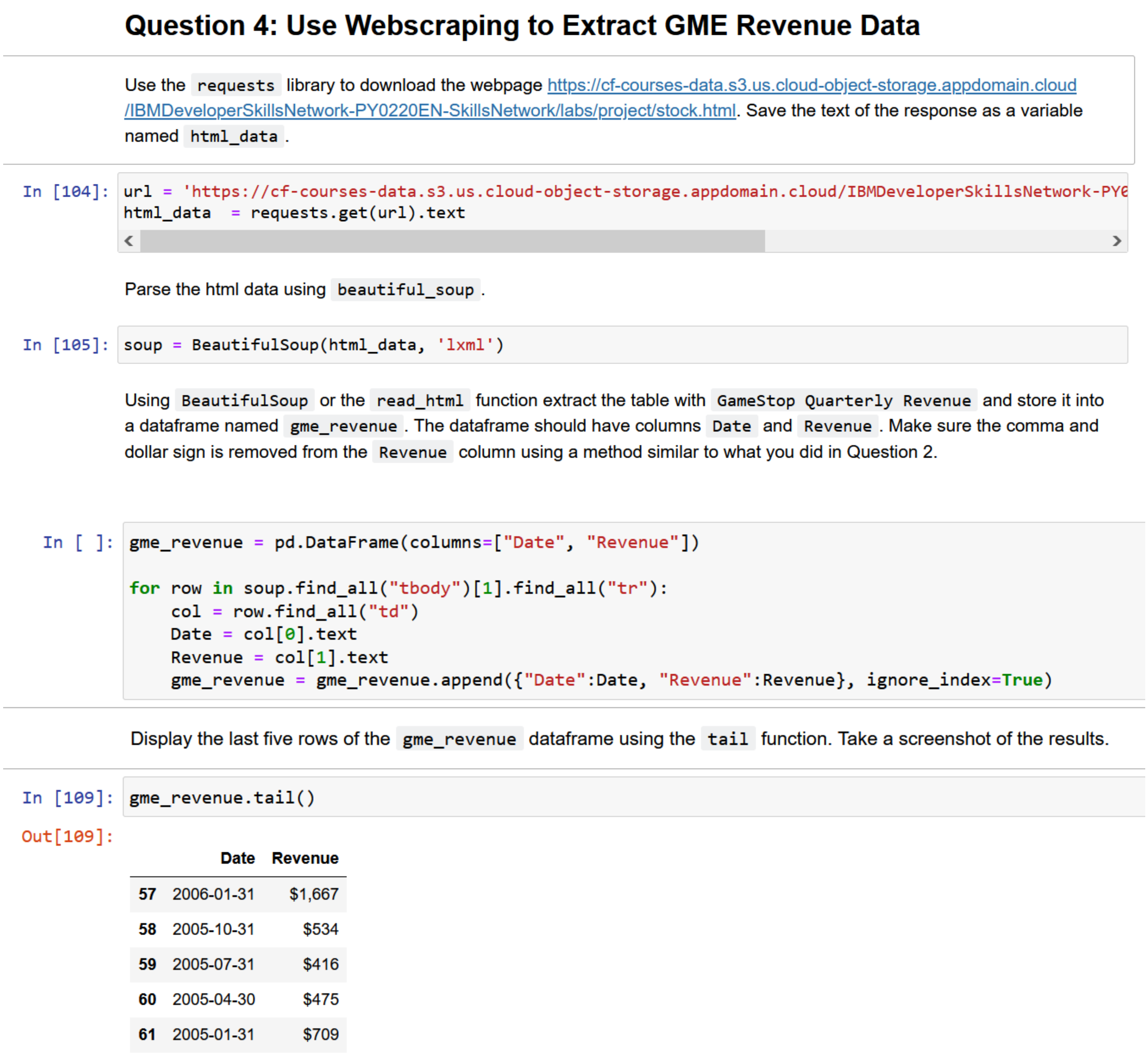
```
In [99]: gamestop = yf.Ticker("GME")
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `gme_data`. Set the `period` parameter to `max` so we get information for the maximum amount of time.

```
In [102]: gme_data = gamestop.history(period='max')
```

Reset the index using the `reset_index(inplace=True)` function on the gme_data DataFrame and display the first five rows of the `gme_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 3 to the results below.

```
In [103]: gme_data.reset_index(inplace=True)
```



Question 4: Use Webscraping to Extract GME Revenue Data

Use the `requests` library to download the webpage https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html. Save the text of the response as a variable named `html_data`.

```
In [104]: url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY
          html_data  = requests.get(url).text
```

Parse the html data using `beautiful_soup`.

```
In [105]: soup = BeautifulSoup(html_data, 'lxml')
```

Using `BeautifulSoup` or the `read_html` function extract the table with `GameStop Quarterly Revenue` and store it into a dataframe named `gme_revenue`. The dataframe should have columns `Date` and `Revenue`. Make sure the comma and dollar sign is removed from the `Revenue` column using a method similar to what you did in Question 2.

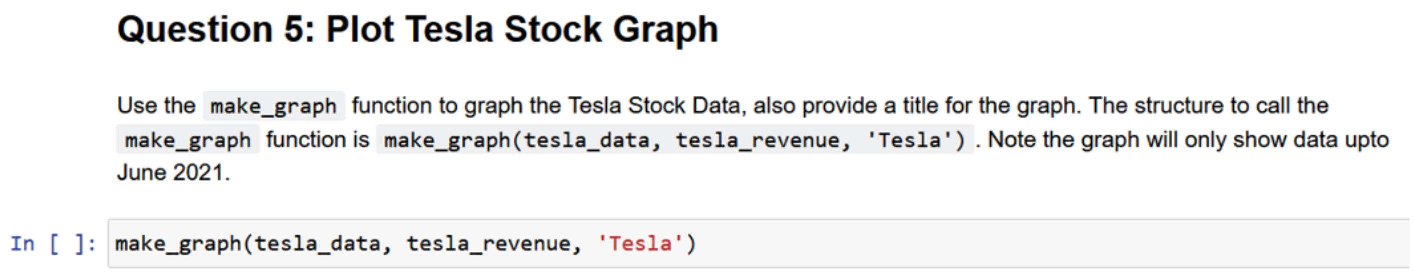
```
In [ ]: gme_revenue = pd.DataFrame(columns=["Date", "Revenue"])

        for row in soup.find_all("tbody")[1].find_all("tr"):
            col = row.find_all("td")
            Date = col[0].text
            Revenue = col[1].text
            gme_revenue = gme_revenue.append({"Date":Date, "Revenue":Revenue}, ignore_index=True)
```

Display the last five rows of the `gme_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
In [109]: gme_revenue.tail()
```

Out[109]:

	Date	Revenue
57	2006-01-31	$1,667
58	2005-10-31	$534
59	2005-07-31	$416
60	2005-04-30	$475
61	2005-01-31	$709

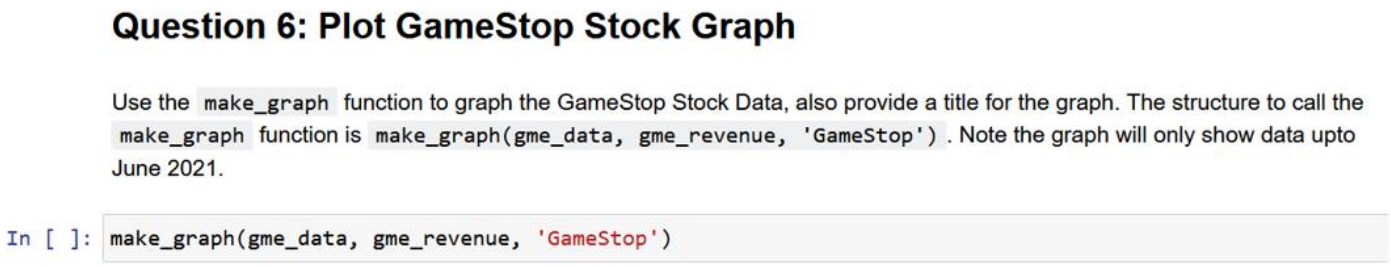


Question 5: Plot Tesla Stock Graph

Use the `make_graph` function to graph the Tesla Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(tesla_data, tesla_revenue, 'Tesla')`. Note the graph will only show data upto June 2021.

```
In [ ]: make_graph(tesla_data, tesla_revenue, 'Tesla')
```

(1) IBM Cloud would not allow me to log in. (2) The Coursera Feature code did not allow me to log into IBM Cloud, either. (3) IBM customer service could not resolve the problem, so I could not graph in IBM Cloud. (4) The Coursera code in this Final Assignment is so bad that it broke my local Jupyter app every time, even after I reinstalled Jupyter. (5) There was no way that I could actually create and display the graph.



Question 6: Plot GameStop Stock Graph

Use the `make_graph` function to graph the GameStop Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(gme_data, gme_revenue, 'GameStop')`. Note the graph will only show data upto June 2021.

```
In [ ]: make_graph(gme_data, gme_revenue, 'GameStop')
```

(1) IBM Cloud would not allow me to log in. (2) The Coursera Feature code did not allow me to log into IBM Cloud, either. (3) IBM customer service could not resolve the problem, so I could not graph in IBM Cloud. (4) The Coursera code in this Final Assignment is so bad that it broke my local Jupyter app every time, even after I reinstalled Jupyter. (5) There was no way that I could actually create and display the graph.